

Response to First Office Action
Docket No. 011.0201.US.UTL

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

- 1 1. (currently amended): A system for efficiently forwarding client
2 requests in a distributed computing environment, comprising:
3 a socket receiving a plurality of non-proxiable requests commonly
4 addressed to an origin server from individual sending clients;
5 a time estimates generator dynamically generating, concurrent to and
6 during processing of each request, time estimates of service availability based on
7 a time-to-idle for sending the requests over each of a plurality of network
8 connections to the origin server; and
9 a network connection manager selecting the network connection to the
10 origin server with a substantially highest service availability and a substantially
11 lowest time-to-idle and forwarding each request to the origin server using the
12 selected network connection.
- 1 2. (currently amended): A system according to Claim 1, further
2 comprising:
3 the network connection manager selecting a network connection not
4 actively sending a request with a zero time-to-idle and not subject to a slow start
5 overhead incurred responsive to flow control imposed by the sending client.
- 1 3. (currently amended): A system according to Claim 2, further
2 comprising:
3 the network connection manager selecting a network connection actively
4 sending a request with a time-to-idle less than the slow start overhead, plus
5 request transfer time if the network connection is pipelined.

Response to First Office Action
Docket No. 011.0201.US.UTL

1 4. (currently amended): A system according to Claim 3, further
2 comprising:
3 the network connection manager selecting a network connection not
4 actively sending a request with a zero time-to-idle and subject to the slow start
5 overhead.

1 5. (currently amended): A system according to Claim 4, further
2 comprising:
3 the network connection manager selecting a network connection actively
4 sending a request with a time-to-idle less than a network connection setup
5 overhead, plus request transfer time if the network connection is pipelined.

1 6. (currently amended): A system according to Claim 5, further
2 comprising:
3 the network connection manager selecting a new network connection in
4 the absence of an existing network connection with a time-to-idle less than the
5 network connection setup overhead.

1 7. (currently amended): A system according to Claim 5, further
2 comprising:
3 the network connection manager selecting an existing network connection
4 with the substantially lowest time-to-idle.

1 8. (currently amended): A system according to Claim 1, wherein the
2 distributed operating environment is TCP/IP-compliant, the system further
3 comprising:
4 the time estimates generator providing time estimates for each network
5 connection comprising at least one of TCP overhead, time-to-idle, idle time, and
6 request transfer time.

Response to First Office Action
Docket No. 011.0201.US.UTL

1 9. (currently amended): A system according to Claim 8, the network
2 connection setup overhead comprises TCP overhead, the system further
3 comprising:
4 the time estimates generator calculating the TCP overhead by adding a
5 three-way handshake overhead to a slow start overhead.

1 10. (currently amended): A system according to Claim 8, further
2 comprising:
3 the time estimates generator calculating the request transfer time by
4 multiplying the size of the request by an average network connection speed for
5 the origin server.

1 11. (currently amended): A system according to Claim 8, further
2 comprising:
3 the time estimates generator calculating the time-to-idle upon each receipt
4 of a request by adding the time-to-idle to the product of an average network
5 connection speed for the origin server multiplied by the sum of the request size
6 and an estimated response size.

1 12. (currently amended): A system according to Claim 8, further
2 comprising:
3 the time estimates generator calculating the time-to-idle upon writing data
4 to a socket by subtracting the time-to-idle from the product of an average network
5 connection speed for the origin server multiplied by the amount of data written.

1 13. (currently amended): A system according to Claim 8, further
2 comprising:
3 the time estimates generator calculating the time-to-idle upon reading data
4 from a socket, prior to header data, by subtracting the time-to-idle from the
5 product of an average network connection speed for the origin server multiplied
6 by the amount of data read.

Response to First Office Action
Docket No. 011.0201.US.UTL

1 14. (original): A system according to Claim 1, further comprising:
2 a proxy configured in a location comprising at least one of local to the
3 sending clients, in the infrastructure of the distributed computing environment,
4 and local to the origin server.

1 15. (currently amended): A method for efficiently forwarding client
2 requests in a distributed computing environment, comprising:
3 receiving a plurality of non-proxiable requests commonly addressed to an
4 origin server from individual sending clients;
5 dynamically generating, concurrent to and during processing of each
6 request, time estimates of service availability based on a time-to-idle for sending
7 the requests over each of a plurality of network connections to the origin server;
8 and
9 selecting the network connection to the origin server with a substantially
10 highest service availability and a substantially lowest time-to-idle and forwarding
11 each request to the origin server using the selected network connection.

1 16. (currently amended): A method according to Claim 15, further
2 comprising:
3 selecting a network connection not actively sending a request with a zero
4 time-to-idle and not subject to a slow start overhead incurred responsive to flow
5 control imposed by the sending client.

1 17. (currently amended): A method according to Claim 16, further
2 comprising:
3 selecting a network connection actively sending a request with a time-to-
4 idle less than the slow start overhead, plus request transfer time if the network
5 connection is pipelined.

1 18. (currently amended): A method according to Claim 17, further
2 comprising:

Response to First Office Action
Docket No. 011.0201.US.UTL

3 selecting a network connection not actively sending a request with a zero
4 time-to-idle and subject to the slow start overhead.

1 19. (currently amended): A method according to Claim 18, further
2 comprising:

3 selecting a network connection actively sending a request with a time-to-
4 idle less than a network connection setup overhead, plus request transfer time if
5 the network connection is pipelined.

1 20. (currently amended): A method according to Claim 19, further
2 comprising:

3 selecting a new network connection in the absence of an existing network
4 connection with a time-to-idle less than the network connection setup overhead.

1 21. (currently amended): A method according to Claim 19, further
2 comprising:

3 selecting an existing network connection with the substantially lowest
4 time-to-idle.

1 22. (currently amended): A method according to Claim 15, wherein the
2 distributed operating environment is TCP/IP-compliant, the method further
3 comprising:

4 providing time estimates for each network connection comprising at least
5 one of TCP overhead, time-to-idle, idle time, and request transfer time.

1 23. (currently amended): A method according to Claim 22, the
2 network connection setup overhead comprises TCP overhead, the method further
3 comprising:

4 calculating the TCP overhead by adding a three-way handshake overhead
5 to a slow start overhead.

1 24. (currently amended): A method according to Claim 22, further
2 comprising:

Response to First Office Action
Docket No. 011.0201.US.UTL

3 calculating the request transfer time by multiplying the size of the request
4 by an average network connection speed for the origin server.

1 25. (currently amended): A method according to Claim 22, further
2 comprising:

3 calculating the time-to-idle upon each receipt of a request by adding the
4 time-to-idle to the product of an average network connection speed for the origin
5 server multiplied by the sum of the request size and an estimated response size.

1 26. (currently amended): A method according to Claim 22, further
2 comprising:

3 calculating the time-to-idle upon writing data to a socket by subtracting
4 the time-to-idle from the product of an average network connection speed for the
5 origin server multiplied by the amount of data written.

1 27. (currently amended): A method according to Claim 22, further
2 comprising:

3 calculating the time-to-idle upon reading data from a socket, prior to
4 header data, by subtracting the time-to-idle from the product of an average
5 network connection speed for the origin server multiplied by the amount of data
6 read.

1 28. (original): A method according to Claim 15, further comprising:
2 providing a proxy configured in a location comprising at least one of local
3 to the sending clients, in the infrastructure of the distributed computing
4 environment, and local to the origin server.

1 29. (original): A computer-readable storage medium holding code for
2 performing the method according to Claim 15.

1 30. (currently amended): A system for efficiently forwarding client
2 requests from a proxy server in a TCP/IP computing environment, comprising:

Response to First Office Action
Docket No. 011.0201.US.UTL

3 means for receiving a plurality of transient requests from individual
4 sending clients, each request being commonly addressed to an origin server;
5 means for dynamically calculating, concurrent to receiving and during
6 processing of each request, time estimates of TCP overhead, slow start overhead,
7 time-to-idle, and request transfer time for sending the requests over each of a
8 plurality of managed network connections to the origin server;
9 means for choosing the managed network connection from, in order of
10 preferred selection, a warm idle network connection, an active network
11 connection with a time-to-idle less than a slow start overhead, a cold idle network
12 connection, an active network connection with a time-to-idle less than a TCP
13 overhead, a new managed network connection, and an existing managed network
14 connection with a smallest time-to-idle; and
15 means for forwarding each request to the origin server over the selected
16 managed network connection.

1 31. (currently amended): A system according to Claim 30, further
2 comprising:

3 means for adding the request transfer time during each active network
4 connection selection if the managed network connection is pipelined.

1 32. (currently amended): A system according to Claim 30, further
2 comprising:

3 means for calculating the TCP overhead by adding a three-way handshake
4 overhead to a slow start overhead;

5 means for calculating the request transfer time by multiplying the size of
6 the request by an average managed network connection speed for the origin
7 server; and

8 means for calculating the time-to-idle, comprising:

9 upon each receipt of a request, means for adding the time-to-idle to
10 the product of an average managed network connection speed for the origin server
11 multiplied by the sum of the request size and an estimated response size;

Response to First Office Action
Docket No. 011.0201.US.UTL

12 upon writing data to a socket, means for subtracting the time-to-
13 idle from the product of an average managed network connection speed for the
14 origin server multiplied by the amount of data written; and
15 upon reading data from a socket, prior to header data, means for
16 subtracting the time-to-idle from the product of an average managed network
17 connection speed for the origin server multiplied by the amount of data read.

1 33. (original): A system according to Claim 30, wherein each transient
2 request is communicated in accordance with HTTP.

1 34. (currently amended): A method for efficiently forwarding client
2 requests from a proxy server in a TCP/IP computing environment, comprising:
3 receiving a plurality of transient requests from individual sending clients
4 into a request queue, each request being commonly addressed to an origin server;
5 dynamically calculating, concurrent to receiving and during processing of
6 each request, time estimates of TCP overhead, slow start overhead, time-to-idle,
7 and request transfer time for sending the requests over each of a plurality of
8 managed network connections to the origin server;
9 choosing the managed network connection from, in order of preferred
10 selection, a warm idle network connection, an active network connection with a
11 time-to-idle less than a slow start overhead, a cold idle network connection, an
12 active network connection with a time-to-idle less than a TCP overhead, a new
13 managed network connection, and an existing managed network connection with
14 a smallest time-to-idle; and
15 forwarding each request to the origin server over the selected managed
16 network connection.

1 35. (currently amended): A method according to Claim 34, further
2 comprising:
3 adding the request transfer time during each active network connection
4 selection if the managed network connection is pipelined.

Response to First Office Action
Docket No. 011.0201.US.UTL

1 36. (currently amended): A method according to Claim 34, further
2 comprising:
3 calculating the TCP overhead by adding a three-way handshake overhead
4 to a slow start overhead;
5 calculating the request transfer time by multiplying the size of the request
6 by an average managed network connection speed for the origin server; and
7 calculating the time-to-idle, comprising:
8 upon each receipt of a request, adding the time-to-idle to the
9 product of an average managed network connection speed for the origin server
10 multiplied by the sum of the request size and an estimated response size;
11 upon writing data to a socket, subtracting the time-to-idle from the
12 product of an average managed network connection speed for the origin server
13 multiplied by the amount of data written; and
14 upon reading data from a socket, prior to header data, subtracting
15 the time-to-idle from the product of an average managed network connection
16 speed for the origin server multiplied by the amount of data read.

1 37. (original): A method according to Claim 34, wherein each transient
2 request is communicated in accordance with HTTP.

1 38. (original): A computer-readable storage medium holding code for
2 performing the method according to Claim 34.